What is claimed is:

1. A method of detecting edges in a compressed video sequence, the compressed video sequence including at least one frame of block encoded video data, the frame of block encoded video data including variable-length codes for transform coefficients of blocks of pixels in the compressed video sequence, the transform coefficients including a respective DC coefficient for each of the blocks of pixels, each respective DC coefficient for at least some of the blocks of pixels being encoded as a respective variable-length code having a length indicating a certain range of differences in DC coefficient values between adjacent ones of the blocks of pixels, wherein the method comprises:

decoding only the length of the respective variable-length code for the respective DC coefficient for each of said at least some of the blocks of pixels in order to produce an indication of whether or not the compressed video sequence includes an edge associated with said each of said at least some of the blocks of pixels; and

performing a code length threshold comparison upon the length of the respective variable-length code for the respective DC coefficient for said each of said at least some of the blocks of pixels for producing at least one respective bit indicating whether or not the compressed video sequence includes an edge associated with said each of said at least some of the blocks of pixels.

2. The method as claimed in claim 1, wherein the compressed video sequence is a color video sequence and there is a respective DC luminance coefficient or

a respective DC C_b chrominance coefficient or a respective DC C_r chrominance coefficient for each of the blocks of pixels depending on a color channel of each of the blocks of pixels, and the method includes:

decoding the length of the respective variable-length code for the respective DC luminance coefficient or DC C_b chrominance coefficient or DC C_r chrominance coefficient of said each of said at least some of the blocks of pixels; and

comparing the decoded length of the respective variable-length code for the respective DC luminance coefficient or DC C_b chrominance coefficient or DC C_r chrominance coefficient of said each of said at least some of the blocks of pixels to at least one length threshold to produce at least one respective bit indicating whether or not the compressed video sequence includes a luminance edge or a C_b chrominance edge or a C_r chrominance edge associated with said each of said at least some of the blocks of pixels.

3. The method as claimed in claim 1, wherein the compressed video sequence is a color video sequence and there is a respective DC luminance coefficient or a respective DC C_b chrominance coefficient or a respective DC C_r chrominance coefficient for each of the blocks of pixels depending on a color channel of each of the blocks of pixels, and the method includes:

decoding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the blocks of pixels;

decoding the length of the respective variable-length code for the respective DC C_b chrominance coefficient of said each of said at least some of the blocks of pixels;

decoding the length of the respective variable-length code for the respective DC C_r chrominance coefficient of said each of said at least some of the blocks of pixels;

combining the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the blocks of pixels with the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the blocks of pixels to produce a combined code length; and

wherein at least one code length threshold is compared to the combined code length for producing at least one respective bit indicating whether or not the compressed video sequence includes an edge associated with said each of said at least some of the blocks of pixels.

4. The method as claimed in claim 3, wherein the combined code length is produced by adding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the blocks of pixels to the sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the blocks of pixels.

5. The method as claimed in claim 1, which includes using a thinning filter for filtering the respective bits indicating whether or not the compressed video sequence includes an edge associated with each of said at least some of the blocks of pixels.

6. The method as claimed in claim 5, wherein the filtering of the respective bits indicating whether or not the compressed video sequence includes an edge associated with said each of said at least some of the blocks of pixels includes comparing the lengths of the respective variable-length codes of the DC coefficients for adjacent blocks of pixels in order to retain indications of edges associated with blocks of pixels having longer variable-length codes for their respective DC coefficients and to exclude indications of edges associated with blocks of pixels having shorter variable-length codes for their respective DC coefficients.

7. The method as claimed in claim 6, wherein an indication of an edge associated with a block of pixels having a shorter variable-length code of the respective DC coefficients for a pair of adjacent blocks of pixels is not excluded upon comparing signs of the respective DC coefficients for the pair of adjacent blocks of pixels and finding that the signs are different.

8. The method as claimed in claim 1, which includes inspecting signs of the respective DC coefficients for said at least some of the blocks of pixels, and based on the signs of the respective DC coefficients for said at least some of the blocks of pixels and based on prediction directions of the respective DC coefficients for said at least some of the blocks of pixels and based on the respective bits indicating whether or not the compressed video sequence includes an edge associated with said at least some of the blocks of pixels, producing a first series of bits indicating whether or not the compressed video sequence includes positive horizontal gradient component edges associated with

said at least some of the blocks of pixels, and producing a second series of bits indicating whether or not the compressed video sequence includes negative horizontal gradient component edges associated with said at least some of the blocks of pixels.

9. The method as claimed in claim 1, which includes inspecting signs of the respective DC coefficients for said at least some of the blocks of pixels, and based on the signs of the respective DC coefficients for said at least some of the blocks of pixels and based on prediction directions of the respective DC coefficients for said at least some of the blocks of pixels and based on the respective bits indicating whether or not the compressed video sequence includes an edge associated with said at least some of the blocks of pixels, producing a first series of bits indicating whether or not the compressed video sequence includes positive vertical gradient component edges associated with said at least some of the blocks of pixels, and producing a second series of bits indicating whether or not the compressed video sequence includes negative vertical gradient component edges associated with said at least some of the blocks of pixels.

10. The method as claimed in claim 1, wherein the transform coefficients include respective horizontal frequency transform coefficients and respective vertical frequency transform coefficients for each block of pixels, and the method includes inspecting a lowest nonzero horizontal frequency transform coefficient and a lowest nonzero vertical frequency transform coefficient for at least one of the blocks of pixels to determine orientation of an edge associated with said at least one of the blocks of pixels.

11. The method as claimed in claim 1, wherein the transform coefficients include respective horizontal frequency transform coefficients and respective vertical frequency transform coefficients for each block of pixels, and the method includes using a lowest nonzero horizontal frequency transform coefficient and a lowest nonzero vertical frequency transform coefficient for at least one of the blocks of pixels for computing an approximate gradient vector of an edge associated with said at least one of the blocks of pixels.

12. A method of detecting edges in a compressed video sequence, the compressed video sequence including at least one I-frame of MPEG video data, the I-frame of MPEG video data including variable-length codes for DCT coefficients of 8x8 pixel blocks in the compressed video sequence, the DCT coefficients including a respective DC coefficient for each of the 8x8 pixel blocks, each respective DC coefficient for at least some of the 8x8 pixel blocks being encoded as a respective variable-length code having a length indicating a certain range of differences in DC coefficient values between adjacent ones of the 8x8 pixel blocks, wherein the method comprises:

decoding only the length of the respective variable-length code for the respective DC coefficient for each of said at least some of the 8x8 pixel blocks in order to produce an indication of whether or not the compressed video sequence includes an edge associated with said each of said at least some of the 8x8 pixel blocks; and

performing a code length threshold comparison upon the length of the respective variable-length code for the respective DC coefficient for said each of said at least some of the 8x8 pixel blocks for producing at least one respective bit indicating whether or not

the compressed video sequence includes an edge associated with said each of said at least some of the 8x8 pixel blocks.

13. The method as claimed in claim 12, wherein the compressed video sequence is a color video sequence and there is a respective DC luminance coefficient or a respective DC C_b chrominance coefficient or a respective DC C_r chrominance coefficient for each of the 8x8 pixel blocks depending on a color channel of each of the 8x8 pixel blocks, and the method includes:

decoding the length of the respective variable-length code for the respective DC luminance coefficient or DC C_b chrominance coefficient or DC C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks; and

comparing the decoded length of the respective variable-length code for the respective DC luminance coefficient or DC C_b chrominance coefficient or DC C_r chrominance coefficient of said each of said at least some 8x8 pixel blocks to at least one length threshold to produce at least one respective bit indicating whether or not the compressed video sequence includes a luminance edge or a C_b chrominance edge or a C_r chrominance edge associated with said each of said at least some of the 8x8 pixel blocks.

14. The method as claimed in claim 12, wherein the compressed video sequence is a color video sequence and there is a respective DC luminance coefficient or a respective DC C_b chrominance coefficient or a respective DC C_r chrominance coefficient for each of the 8x8 pixel blocks depending on a color channel of each of the 8x8 pixel blocks, and the method includes:

decoding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks;

decoding the length of the respective variable-length code for the respective DC C_b chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;

decoding the length of the respective variable-length code for the respective DC C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;

combining the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks with the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the 8x8 pixel blocks to produce a combined code length; and

wherein at least one code length threshold is compared to the combined code length for producing at least one respective bit indicating whether or not the compressed video sequence includes an edge associated with said each of said at least some of the 8x8 pixel blocks.

15. The method as claimed in claim 14, wherein the combined code length is produced by adding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks to the sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the 8x8 pixel blocks.

16. The method as claimed in claim 12, which includes using a thinning filter for filtering the respective bits indicating whether or not the compressed video sequence includes an edge associated with each of said at least some of the 8x8 pixel blocks.

17. The method as claimed in claim 16, wherein the filtering of the respective bits indicating whether or not the compressed video sequence includes an edge associated with said each of said at least some of the 8x8 pixel blocks includes comparing the lengths of the respective variable-length codes of the DC coefficients for adjacent 8x8 pixel blocks in order to retain indications of edges associated with 8x8 pixel blocks having longer variable-length codes for their respective DC coefficients and to exclude indications of edges associated with 8x8 pixel blocks having shorter variable-length codes for their respective DC coefficients.

18. The method as claimed in claim 17, wherein an indication of an edge associated with a block of pixels having a shorter variable-length code of the respective DC coefficients for a pair of adjacent 8x8 pixel blocks is not excluded upon comparing signs of the respective DC coefficients for the pair of adjacent 8x8 pixel blocks and finding that the signs are different.

19. The method as claimed in claim 12, which includes inspecting signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on prediction directions of the respective DC coefficients for said at least some of

the 8x8 pixel blocks and based on the respective bits indicating whether or not the compressed video sequence includes an edge associated with said at least some of the 8x8 pixel blocks, producing a first series of bits indicating whether or not the compressed video sequence includes positive horizontal gradient component edges associated with said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating whether or not the compressed video sequence includes negative horizontal gradient component edges associated with said at least some of the 8x8 pixel blocks.

The method as claimed in claim 11, which includes inspecting signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on prediction directions of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on the respective bits indicating whether or not the compressed video sequence includes an edge associated with said at least some of the 8x8 pixel blocks, producing a first series of bits indicating whether or not the compressed video sequence includes positive vertical gradient component edges associated with said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating whether or not the compressed video sequence includes negative vertical gradient component edges associated with said at least some of the 8x8 pixel blocks.

20. The method as claimed in claim 12, wherein the DCT coefficients include respective horizontal frequency DCT coefficients and respective vertical frequency DCT coefficients for each of the 8x8 pixel blocks, and the method includes inspecting a lowest nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency

DCT coefficient for at least one of the 8x8 pixel blocks to determine orientation of an edge associated with said at least one of the 8x8 pixel blocks.

21. The method as claimed in claim 12, wherein the DCT coefficients include respective horizontal frequency DCT coefficients and respective vertical frequency DCT coefficients for each of the 8x8 pixel blocks, and the method includes using a lowest nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency DCT coefficient for at least one of the 8x8 pixel blocks for computing an approximate gradient vector of an edge associated with said at least one of the 8x8 pixel blocks.

22. A method of detecting a scene change between I-frames of MPEG video data, said method comprising:

a) detecting edges in images represented by the I-frames by decoding lengths of variable-length codes for DCT DC coefficients of 8x8 pixel blocks in the I-frames and performing code length threshold comparisons upon the decoded code lengths to produce respective edge indications for each of the I-frames; and

b) comparing the edge indications between the I-frames in order to signal a scene change when there is a significant change in the edge indications between the I-frames.

23. The method as claimed in claim 22, wherein the detecting of edges in the images includes producing a frame of bits for at least one of the I-frames, the frame of bits including at least one respective bit for each of the 8x8 pixel blocks in said at least

one of the I-frames, and storing the frame of bits for said at least one of the I-frames in a frame buffer, and

wherein the comparing of the edge indications between the I-frames includes accessing the frame of bits in the frame buffer for comparing the edge indications for an I-frame following said at least one of the I-frames to the edge indications for said at least one of the I-frames.

24. The method as claimed in claim 22, wherein the comparing of the edge indications between the I-frames includes extracting features from the edge indications for each of the I-frames and comparing features extracted between the I-frames.

25. The method as claimed in claim 22, wherein the comparing of the edge indications between the I-frames includes computing characteristics of the edge indications for each of the I-frames and comparing the characteristics of the edge indications between the I-frames.

26. The method as claimed in claim 22, wherein the 8x8 pixel blocks each have a respective DC luminance coefficient or a respective DC C_b chrominance coefficient or a respective DC C_r chrominance coefficient depending on a color channel of the 8x8 pixel block, and the method includes:

decoding the length of the respective variable-length code for the respective DC luminance coefficient of each of at least some of the 8x8 pixel blocks;

decoding the length of the respective variable-length code for the respective DC C_b chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;

decoding the length of the respective variable-length code for the respective DC C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;

combining the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks with the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the 8x8 pixel blocks to produce a combined code length; and

comparing at least one code length threshold to the combined code length for producing at least one respective bit providing an edge indication for said each of said at least some of the 8x8 pixel blocks.

27. The method as claimed in claim 26, wherein the combined code length is produced by adding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks to the sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the 8x8 pixel blocks.

28. The method as claimed in claim 22, which includes using a thinning filter for filtering the respective edge indications.

29. The method as claimed in claim 28, wherein the filtering of the respective edge indications includes comparing the lengths of the respective variable-length codes of the DC coefficients for adjacent 8x8 pixel blocks in order to retain indications of edges associated with 8x8 pixel blocks having longer variable-length codes for their respective DC coefficients and to exclude indications of edges associated with 8x8 pixel blocks having shorter variable-length codes for their respective DC coefficients.

30. The method as claimed in claim 29, wherein an indication of an edge associated with a block of pixels having a shorter variable-length code of the respective DC coefficients for a pair of adjacent 8x8 pixel blocks is not excluded upon comparing signs of the respective DC coefficients for the pair of adjacent 8x8 pixel blocks and finding that the signs are different.

31. The method as claimed in claim 22, which includes inspecting signs of the respective DC coefficients for at least some of the 8x8 pixel blocks, and based on the signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on prediction directions of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on respective bits indicating whether or not an edge is associated with said at least some of the 8x8 pixel blocks, producing a first series of bits indicating whether or not positive horizontal gradient component edges are associated with said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating whether or not negative horizontal gradient component edges are associated with said at least some of the 8x8 pixel blocks.

32. The method as claimed in claim 22, which includes inspecting signs of the respective DC coefficients for at least some of the 8x8 pixel blocks, and based on the signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on prediction directions of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on respective bits indicating whether or not an edge is associated with said at least some of the 8x8 pixel blocks, producing a first series of bits indicating whether or not positive vertical gradient component edges are associated with said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating whether or not negative vertical gradient component edges are associated with said at least some of the 8x8 pixel blocks.

33. The method as claimed in claim 22, wherein the DCT coefficients include respective horizontal frequency DCT coefficients and respective vertical frequency DCT coefficients for each of the 8x8 pixel blocks, and the method includes inspecting a lowest nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency DCT coefficient for at least one of the 8x8 pixel blocks to determine orientation of an edge associated with said at least one of the 8x8 pixel blocks.

34. The method as claimed in claim 22, wherein the DCT coefficients include respective horizontal frequency DCT coefficients and respective vertical frequency DCT coefficients for each of the 8x8 pixel blocks, and the method includes using a lowest nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency

DCT coefficient for at least one of the 8x8 pixel blocks for computing an approximate gradient vector of an edge associated with said at least one of the 8x8 pixel blocks.

35. A method of detecting a scene change between I-frames of MPEG video data, each of the I-frames including a series of 8x8 pixel blocks, said method comprising:

a) detecting edges in images represented by the I-frames to produce a series of respective bits indicating whether or not an edge is associated with at least some of the 8x8 pixel blocks;

b) filtering the series of the respective bits indicating whether or not an edge is associated with said at least some of the 8x8 pixel blocks with a thinning filter in order to produce a filtered series of respective bits including more significant edge indications and excluding less significant edge indications; and

c) operating a digital processor to process the filtered series of respective bits in order to signal a scene change when there is a significant change in features between the I-frames.

36. The method as claimed in claim 35, which includes storing a frame of bits of the filtered series of respective bits for at least one of the I-frames in a frame buffer, and wherein the digital processor accesses the frame of bits in the frame buffer for comparing edge indications for an I-frame following said at least one of the I-frames to edge indications for said at least one of the I-frames.

37. The method as claimed in claim 35, wherein the processor extracts features from the filtered series of respective bits for each of the I-frames and compares features extracted from at least one of the I-frames to features extracted from an I-frame following said at least one of the I-frames.

38. The method as claimed in claim 35, wherein the processor computes characteristics of the filtered series of respective bits for each of the I-frames and compares the characteristics of the filtered series of respective bits for at least one of the I-frames to the characteristics of the filtered series of respective bits for an I-frame following said at least one of the I-frames.

39. The method as claimed in claim 35, wherein the detecting of edges in images represented by the I-frames includes decoding lengths of variable-length codes of DCT DC transform coefficients of each of said at least some of the 8x8 pixel blocks and performing a length threshold comparison upon the decoded lengths of the variable-length codes to produce the series of respective bits indicating whether or not an edge is associated with said at least some of the 8x8 pixel blocks.

40. The method as claimed in claim 39, wherein the 8x8 pixel blocks each have a respective DC luminance coefficient or a respective DC C_b chrominance coefficient or a respective DC C_r chrominance coefficient depending on a color channel of the 8x8 pixel block, and the method includes:

decoding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks;

decoding the length of the respective variable-length code for the respective DC C_b chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;

decoding the length of the respective variable-length code for the respective DC C_r chrominance coefficient of said each of said at least some of the 8x8 pixel blocks;

combining the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks with the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the 8x8 pixel blocks to produce a combined code length; and

comparing at least one code length threshold to the combined code length for producing at least one respective bit providing an edge indication for said each of said at least some of the 8x8 pixel blocks.

41. The method as claimed in claim 40, wherein the combined code length is produced by adding the length of the respective variable-length code for the respective DC luminance coefficient of said each of said at least some of the 8x8 pixel blocks to the sum of the lengths of the respective variable-length codes for the respective DC C_b and C_r chrominance coefficients of said each of said at least some of the 8x8 pixel blocks.

42. The method as claimed in claim 39, wherein the filtering of the series of the respective bits indicating whether or not an edge is associated with said at least some

of the 8x8 pixel blocks includes comparing the decoded lengths of the variable-length codes for adjacent 8x8 pixel blocks in order to retain edge indications for 8x8 pixel blocks having longer decoded code lengths and to exclude edge indications for 8x8 pixel blocks having shorter decoded code lengths.

43. The method as claimed in claim 42, wherein an indication of an edge associated with an 8x8 pixel block having a shorter variable-length code of the respective DC coefficients for a pair of adjacent 8x8 pixel blocks is not excluded upon comparing signs of the respective DC coefficients for the pair of adjacent 8x8 pixel blocks and finding that the signs are different.

44. The method as claimed in claim 35, which includes inspecting signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on prediction directions of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on the respective bits indicating whether or not an edge is associated with said at least some of the 8x8 pixel blocks, producing a first series of bits indicating whether or not positive horizontal gradient component edges are associated with said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating whether or not negative horizontal gradient component edges are associated with said at least some of the 8x8 pixel blocks.

45. The method as claimed in claim 35, which includes inspecting signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks, and based on the signs of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on prediction directions of the respective DC coefficients for said at least some of the 8x8 pixel blocks and based on the respective bits indicating whether or not an edge is associated with said at least some of the 8x8 pixel blocks, producing a first series of bits indicating whether or not positive vertical gradient component edges are associated with said at least some of the 8x8 pixel blocks, and producing a second series of bits indicating whether or not negative vertical gradient component edges are associated with said at least some of the 8x8 pixel blocks.

46. The method as claimed in claim 35, which includes inspecting a lowest nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency DCT coefficient for at least one of the 8x8 pixel blocks to determine orientation of an edge associated with said at least one of the 8x8 pixel blocks.

47. The method as claimed in claim 35, which includes using a lowest nonzero horizontal frequency DCT coefficient and a lowest nonzero vertical frequency DCT coefficient for at least one of the 8x8 pixel blocks for computing an approximate gradient vector of an edge associated with said at least one of the 8x8 pixel blocks.